Software Based Motion Picture Coding

A project report submitted
in partial fulfilment of the requirements
for the DIIT(Diploma in Indian Institute of Technology)

by

C. L. SEBASTIAN

to the

DEPARTMENT OF ELECTRICAL ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY KANPUR

MAY 1994

CERTIFICATE

This is to certify that the thesis work entitled *Software based motion picture coder* by C L SEBASTIAN has been carried out under my supervision and the same has not been submitted else where for a P.G.diploma

Dr Govind Sharma
Associate Professor
Department of Electrical Engineering
Indian Institute of Technology
Kanpur - 208016
INDIA

May 1994

EE-1994-D-SEB-SOF

ACKNOWLEDGEMENTS

First of all I would like to express my deep sense of gratitude to my thesis supervisor, Dr Govind Sharma for suggesting me this exciting and interesting project, and for his invaluable guidance and constant help I have benefited significantly from his experience and expertise particularly during this project work and the entire course in general

I would also like to thank Prashanthan, for his invaluable suggestions. He has generously donated many days and hours of his valuable time.

I am thankful to Doordarshan for sponsoring me for this course.

I am thankful to many friends, friends of friends for their help Those whom I would particularly like to acknowledge are Deepak Murthy, Viswanath, Prabhucharan, Balwinder, Ramesh, Ilangoven and Anjanikumar I am thankful to all my Doordarshan colleagues for making my stay at I I T memorable

No words to thank my parents for providing me with primary education, during their difficult times

Last but not least I thank my wife and daughter for tolerating the long separation and providing moral support through their letters

C L SEBASTIAN

To my
Wife
and
Daughter

Abstract

Video in its digital form has huge data rate. Hence handling of digital video either for storage or for transmission has become a challenge. The way out is to efficiently compress them.

In its earliest form, digital video compression was applied to all the image data. The resulting images had poor definition and jerky movements accompanied by blurring. This might be acceptable for static images, but it is scarcely acceptable for television, especially for fast moving action such as sports.

So the researchers were forced to invent a means of selective compression. Since the consecutive video frames are very similar in nature, a lot of compression is possible. In this, the parts of the overall image which did not change from frame to frame (such as the background, be it sky or wallpaper) could be subjected to quite severe compression, while fast moving picture elements were compressed to a much lesser extent. With the processing power available today, it has become a reality to perform selective compression or coding in real time.

This project is one such selective compression technique called as "Software based moving picture coder" (SBMPC). The heart of the system is the modified block truncation coder, though multiresolution sampling technique is used. Interframe coding technique is used to exploit the temporal correlation between frames. DPCM technique is used to detect the motion between the frames. The encoder and the decoder is implemented on PC/AT 386

Contents

Chapter 1

Introduction

While developing an image processing system for practical applications, the first important step is to identify clearly the overall objective. In applications for which the images are processed for human viewers like in television, the properties of human visual perception can be taken into account

The main objective of image coding is to represent an image with as few bits as possible at the same time preserving the quality and intelligibility required for the given applications. The levels of quality and intelligibility to be preserved, depends on the type of application. For example in application such as storage of image data from space programs, we may want to preserve all the information, while in application such as digital television, though high quality is important, some information in the original data may be destroyed, as long as the displayed image on the T.V. screen is acceptable to the human viewer.

Image coding has basically two applications. The first one is reduction of channel bandwidth for transmission *eg. Digital Television.* The other application is reduction of storage requirements *eg. Digital VCR.* For transmission, compression techniques are more complicated because of realtime requirements. but for storage applications the compression technique can be less stringent because much of the preprocessing can be done off line.

Image data compression methods can be classified into two basically different categories. In the first category are those methods which exploit *Redundancy* in the data. Redundancy is a charactaristics which is related to predictability, randomness, smoothness, etc., in the data. For example an image of constant grey levels is fully predictable once the grey level of the first pixel is known On the other hand, a white noise random field is totally unpredictable and every pixel has to be stored to reproduce the image. Thus many data compression algorithms attempt to represent a given sampled image array $x(i,j)$, by another array $u(i,j)$, which has no redundancy and such that $x(i,j)$ can be determined uniquely from $u(i,j)$. The raw data rate of $u(i,j)$ then

determines the data rate $x(i,j)$. Often such a process results in some compression but with an accompanying distortion in the reproduced array $x(i,j)$. Efficient compression techniques tend to minimize this distortion.

There are various approaches to video coding. A broad classification is as follows.

- Predictive coding.
- Transform coding.
- Interpolative coding.

In *predictive coding*, also known as DPCM, the sample to be encoded is predicted from the encoded values of the previously transmitted samples and only the prediction error is quantized for transmission. Such an approach can be made adaptive either by changing the prediction or quantization or by not transmitting the prediction error whenever it is below a certain threshold It is also possible to delay the encoding of a sample by observing the trend of the signal as indicated by certain subsequent samples.

In *transform coding*, an alternative representation of the signal is made first, by taking linear combinations of samples in a block of data called as coefficients and then quantizing the selected coefficients for transmission The primary purpose of transformation is to convert statistically dependant pixels into independant coefficients. Several transformation have been used, but the most popular one is *discrete cosine transform* (DCT). For natural pictures, DCT leads to an energy concentration in only a few coefficients containing the main part of the information. The coding takes advantage of the statistical dependancies of pixels for redundancy reduction.

The next class of coding techniques, the *interpolative coding*, attempts to send certain samples to the reciever and interpolate all the rest. Again adaptation can be built in by varying the criterion for the selection of the samples to be sent and the strategy for interpolating the remaining samples.

The coding can also be classified into *intraframe and interframe* coding In intraframe coding only the first frame is coded, thus only the spatial correlation is exploited Interframe coding is used for sequences of similar frames, including those containing moving objects. Intraframe coding removes only the spatial redundancy within a picture, where as interframe coding removes the temporal redundancy between the pictures

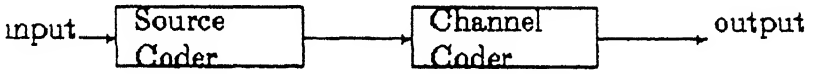
Some of the standard video coding techniques such as H.261 and MPEG have been designed to compress video data by both reducing the spatial correlation through the transform coding (DCT), and the temporal correlation through the predictive coding (DPCM). One of the major problems of these techniques are the high computational complexity which inhibits these techniques for real time applications The SBMPC is a software based video coder and it is relatively a fast algorithm compared to transform

coding. The heart of the system is the modified block truncation coder. The SBMPC is implemented using this technique with 64×64 frame size.

Chapter 2

Preliminary reviews

In practice transmission channels are frequently prone to errors and when the signal is represented more efficiently the effect of an error becomes far more serious. Consequently, it is often necessary to add a controlled form of redundancy back into the signal in the form of channel encoding in order to reduce the impact of trnsmission errors.



The typical configuration then, is shown in Fig. 2 with the coding broken down into *source encoding* in which redundancy is removed from the signal for the purpose of achieving a more efficient representation, and *channel encoding* where redundancy is reinserted into the signal in order to obtain better channel-error performance. Offcourse, the increase in bit rate resulting from the channel coding stage should be significantly less than the decrease in bit rate resulting from the source encoding operation in order to realise a saving.

In practice the application of picture coding to transmission channels is an economic tradeoff in system design, picture quality, bit rate and error performance.

Table 1 shows the classification of the approaches that have been used for picture coding

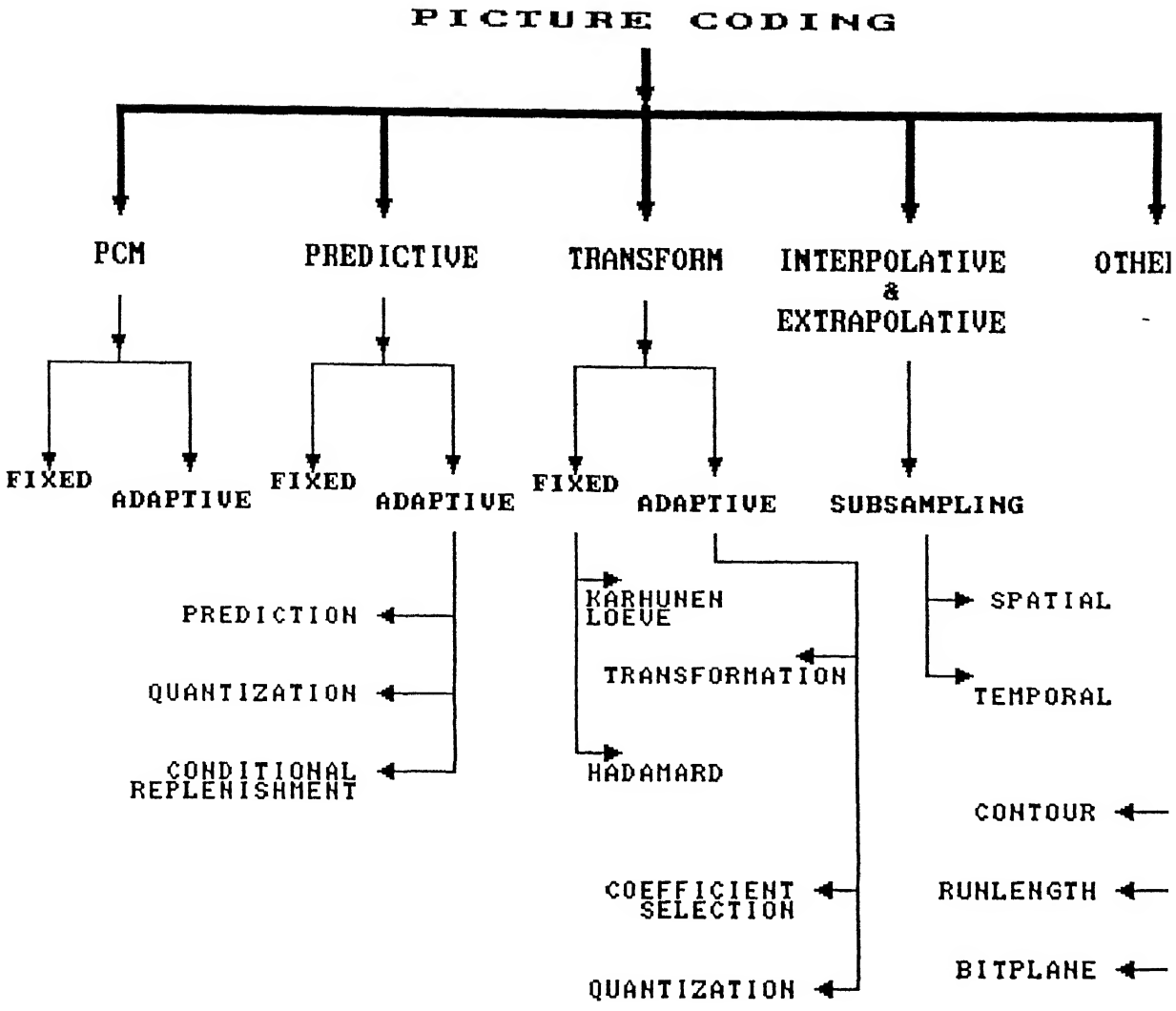


TABLE 1: CLASSIFICATION OF PICTURE CODING

2.1 Pulse Code Modulation

Pulse code modulation is nothing more than a time discrete, amplitude discrete representation of the signal. It is used as a digitizing scheme for purposes of transmission and also for digitizing before application of other, more sophisticated coding techniques.

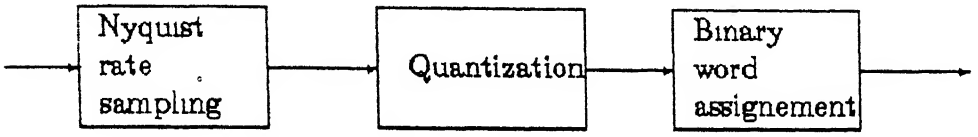


Figure 2.1: Pulse Code Modulation

The basic PCM consist of sampling the waveform (usually at the nyquist rate) and quantizing each sample using N levels. Each level is represented by a binary word containing B bits N is usually taken to be a power of two. In the decoder, these binary words are converted to discrete amplitude levels, and then the time sequence of the amplitude levels are low pass filtered. Basic PCM affords a simplicity uncommon to most other coders, but suffers from inefficiency since it does not use redundancy present in the picture signal. PCM coding systems, in general, require about 128 to 256 levels (7 to 8 bits) for good quality pictures under most viewing conditions.

2.2 Predictive coding

In predictive coding, a prediction of the sample to be encoded is made from previously coded information that has been transmitted. The error resulting from the subtraction of the prediction from the actual value of the sample is quantized into a set of discrete amplitude levels. These levels are represented as binary words of either fixed or variable word length and sent to the channel coder for transmission. Thus the predictive coder has two basic components:

- Predictor.
- quantizer

Depending upon the number of levels of the quantizer, a distinction is often made between Delta modulation (DM) (N=2) and DPCM, which has N greater than 2.

Although DM has been extensively used in encoding other waveforms (eg. Speech), it is not found great use in encoding of pictures, due perhaps to the high sampling rates required.

In its simplest form, DPCM uses the coded value of the horizontally previous pixel as the prediction. However, more sophisticated predicters use the previous line (2-D predictor) as well as previous frame of information (Interframe predictor). It is the design of the predictors which optimizes the efficiency of the coder.

DPCM schemes achieve compression to a large extent, by not quantizing the prediction error as finely as the original signal itself. Hence the compression in the DPCM entirely depends upon quantization. Several methods of optimizing quantizers have been studied, but quantizer design still remains an art. Most of the work on systematic procedures for quantizer optimization has been for the previous element DPCM coding, in which approximate horizontal slope of the input signal is quantized.

Distortions of DPCM image coding are granular noise (grainy structure in uniform regions), contouring patterns, slope overload near edges (visible losses in edge) and edge busyness Reducing granular noise requires fine quantizer levels, where as attenuation of slope overload can be accomplished by large quantization steps. These conflicting requirements compel the quantizer to be designed based on art rather than on the basis of a mathematical model.

Quantizers matched to the human visual sensitivity have been designed. These quantizers mask the DPCM distortions and generate visually acceptable images. Edgy busyness is defined as a noisy edge when the prediction scheme is unable to reconstruct the precise continuity at an edge. In interframe coding this defect appears as a blurred moving edge. When the number of quantization levels are too small, the images have layered effects, generating contouring patterns. In predictive coding a noisy channel can cause error propagation, hence DPCM technique for coding is suicidal against channel errors.

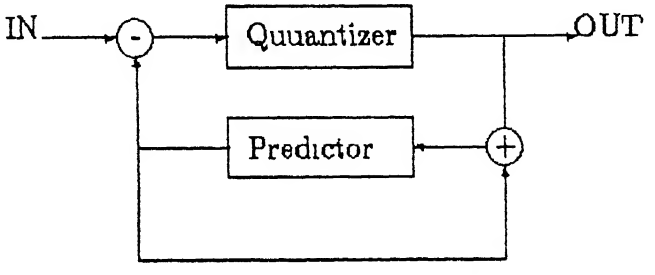


Figure 2 2. Predictive Coding

2.3 Interpolative coding

In interpolative coding picture elements are subsampled and only the subsampled elements are transmitted and the left out elements are interpolated at the reciever. In general interpolative coding can be classified as fixed interpolation and adaptive interpolation

In fixed interpolative methods a fixed set of picture elements are selected for transmission and the rest are interpolated. Various types of samples can be chosen for transmission As examples one may transmit every alternate sample, one sample out of four, every alternate scan line, every alternate frame The picture quality that results is the function of the number and type of samples that are dropped and the method of interpolation. Most interpolation methods have used weighted averages either using straight lines or higher degree polynomials. It appears that interpolation using straight lines is quiete effective and not much is gained by interpolation using polynomials of higher degree. Another method of interpolation, drops alternate fields and attempts to construct them by making movement of edges temporarily continuous, i.e., placing the edges in the dropped field at places dictated by their uniform motion between the adjacent transmitted field. Such technique may be useful, but in practice it has been found to be rather difficult to implement since it requires defination of edges and their motion.

Adaptive interpolative coding consists of choosing certain points for transmission, constructing the interpolation of non transmitted pixels and evaluating the interpolation error. If the error is below a certain threshold, then fewer points are chosen for transmission. The fewest number of points that are needed to keep the interpolation error below the threshold is normally the desired charactaristics.

Performance of interpolative coders depends upon the technique used for transmitting samples and for interpolating. In terms of number of bits required for a given quality, interpolative coders show an improvement over nonadaptive DPCM, but they appear to be inferior to both adaptive DPCM and transform techniques

2.4 Transform coding

In trnsform coding, a picture is divided into subpictures and each subpictures are transformed into a set of more independant coefficients. The coefficients are then quantized and coded for transmission. At the reciever the recieved bits are decoded into transform coefficients. An inverse transformation is applied to recover intensities of picture elements Much of the compression is a result of dropping coefficients from the transmission that are small and coarsely quantizing the others as required by the picture quality The important parameter that determine the performance of the transform

coder are size and shape of subpictures, type of transformation used, selection of the coefficients to be transmitted and quantization of them, and the bit assigner which assigns a binary word for each of the quantizer outputs

The transformation widely used for image coding is the DCT. DCT has the energy packing capabilities. The development of various fast algorithms for efficient implementation of DCT further contributed to its popularity.

The two important properties of DCT are *orthogonality* and *seperability*. Orthogonality implies that the energy or information of a signal is preserved under transformation (mapping into the DCT domain). Seperability implies that a multidimensional DCT can be implemented by a series of one-dimensional transforms The benifit of this property is that the fast algorithms developed for the one-dimensional DCT can be directly extended to multidimensional transforms.

As an exercise I implemented DCT for video compression using following DCT transforms.

$$C_x[k_1,k_2] = \sum_{n_1=0}^{N_1-1}\sum_{n_2=0}^{N_2-1} 4x[n_1,n_2]cos\frac{\pi}{2N_1}k_1(2n_1+1)cos\frac{\pi}{2N_2}k_2(2n_2+1)$$

$$0 \leq k_1 \leq N_1 - 1$$

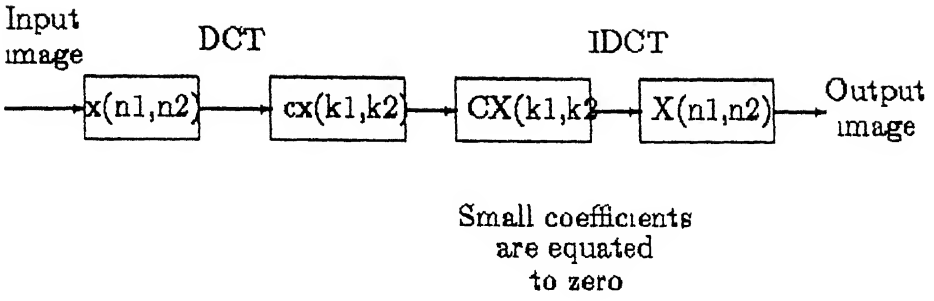
$$0 \leq k_2 \leq N_2 - 1$$

$$x[n_1,n_2] = \frac{1}{N_1N_2}\sum_{k_1=0}^{N_1-1}\sum_{k_2=0}^{N_2-1} W_1[k_1]W_2[k_2]C_x[k_1,k_2]cos\frac{\pi}{2N_1}k_1(2n_1+1)cos\frac{\pi}{2N_2}k_2(2n_2+1)$$

$$W_1[k_1] = \begin{cases} \frac{1}{2} & , k_1 = 0 \\ 1 & , 1 \leq k_1 \leq N_1 - 1 \end{cases}$$

$$W_2[k_2] = \begin{cases} \frac{1}{2} & , k_2 = 0 \\ 1 & , 1 \leq k_2 \leq N_2 - 1 \end{cases}$$

A 32×32 image was taken and using the above equations and diagram the image was compressed. The main disadvantage was high computation time But the compressed image was of good quality.



The block diagram shown above was the set up used to implement DCT for video compression The discrete cosine transform of the image was first taken using the equations mentioned above. The coefficients of the DCT was observed. All the coefficients which were small were equated zero. A threshold was set, and all the values less than that threshold were equated to zero. Different threshold levels were tried.

After making the small coefficients zero the inverse discrete cosine transform was applied and the image was displayed By varying the threshold i.e.,by changing the compression ratio the image quality was observed

Chapter 3

Software Based Motion Picture Coder

The heart of the software based motion picture coder is the modified block truncation coder and the multiresolution subsampling technique. In the next section, the block truncation coder is described and the reason to modify the block truncation coder is explored, so that it can meet the requirements of software based motion picture coder In subsequent sections the modified block truncation coder, the SBMPC coder and decoder and the way it was implemented will be described.

3.1 Block truncation coding algorithm

In block truncation coding (BTC) the image is first divided into $n \times n$ blocks. The blocks are coded individually, each into a two level signal. The levels for each block are chosen such that the first two sample moments are presevered.

Let $m = n^2$ and let $x_1, x_2, \ldots x_m$ be the values of the pixels in a block of the original picture. Then the first and the second sample moments and the sample variance are, respectively. $X_1 = \frac{1}{m}\sum_{i=1}^{m} x_i$, and $X_2 = \frac{1}{m}\sum_{i=1}^{m} x_i^2$ and $\sigma^2 = X_2 - X_1^2$. As with the design of any, one bit quantizer we find a threshold, X_{th} and two output levels a and b such that if $X_i \geq X_{th}$ $output = b$ and if $X_i < X_{th}$ $output = a$ for $i = 1, 2, \ldots, m$. For the quantizer, we set $X_{th} = X_1$. The output levels a and b are found by solving the following equations. Let q = number of x_i's greater than X_{th}.

Then in order to preserve $X_1 and X_2$, $mX_1 = (m-q)a+qb$ and $mX_2 = (m-q)a^2+qb^2$ Solving for a and b

$$a = X_1 - \sigma\sqrt{[\frac{q}{m-q}]},$$

$$b = X_1 + \sigma\sqrt{[\frac{m-q}{q}]}.$$

Each block is then described by the values of X_1, σ, and an $n \times n$ bit plane consisting of 1's and 0's indicating whether pixels are above or below X_{th}. The reciever reconstructs the image block by calculating a and b and then assigning these values to pixels in accordance with the code in the bit plane.

Illustration of block truncation coding for a 4×4 picture block

- The image is first divided into small non overlapping blocks of 4×4.
- The first and second sample moments are computed.
- A bit plane is constructed such that each pixel location is coded as a "one" or a "zero" depending on whether that pixel is greater than X_1
- The bit plane, X_1 and σ are sent to the reciever.
- The picture block is reconstructed such that X_1 and σ are preserved That is, pixels in the bit plane that are "0" are set to "a" and the "1"'s are set to "b"

For example, suppose a 4×4 picture block is given by thje following

$$X_{i,j} = \begin{bmatrix} 121 & 114 & 56 & 47 \\ 37 & 200 & 247 & 255 \\ 16 & 0 & 12 & 169 \\ 43 & 5 & 7 & 251 \end{bmatrix}$$

Hence, $X_1 = 98.75$, $\sigma = 92.95$, $q = 7$, and $a = 16.7 \simeq 17$, $b = 204\,2 \simeq 204$. The bit plane is

$$\begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

The reconstructed block becomes

$$\begin{bmatrix} 204 & 204 & 17 & 17 \\ 17 & 204 & 204 & 204 \\ 17 & 17 & 17 & 204 \\ 17 & 17 & 17 & 204 \end{bmatrix}$$

and the Sample mean and Variance are preserved.

Because the calculations are relatively simple and storage of data small, BTC is fairly easy to implement. BTC was implemented using 32×32 image. The decoded image quality was good.

The advantages of BTC are listed below, and these are the reasons BTC was chosen as the basic kernel of SBMPC.

- The coding speed of BTC is very fast.
- The BTC is an edge preserving algorithm.
- BTC retains important visual features, while discarding details which are not likely to be noticable by a human observer.

Although the BTC has the benefits described above, there are some inborn defects existed in BTC and these are the reasons that why BTC cannot be used directly in SBMPC.

- The compression ratio of BTC is not high enough to meet the bit rate requirement of a video coding system.
- Although the coding speed of BTC is fast, it is still not fast enough to meet the real time requirement of SBMPC system.

The modified BTC which will be described in the next section is designed to overcome the limitations of BTC mentioned above.

3.2 Modified BTC

The modification incorporated in modified BTC is mainly subsampling Like in Block truncation coding, the input image is first segmented into blocks Block sizes of 4×4, 8×8 or 16×16 are generally utilized in image coding.

These small block sizes enables us to introduce adaptive features based on activity or detail in the block. Although there may be some correlation between neighbouring blocks, this is too insignificant to warrant large block size beyond 16×16.

In modified BTC block size of 8×8 is used. After the image is segmented into blocks, the pixels within the blocks are subsampled. The subsampling process increases compression ratio as well as increases the computation speed. Depending upon the requirement of computation speed, compression ratio and quality of picture, the subsampling rate can be chosen. Also the subsampling rate can be made adaptive according to the changes in the picture frame. Furthermore, a simplified algorithm is used for computing the quantizers threshold and the two reconstruction levels for each

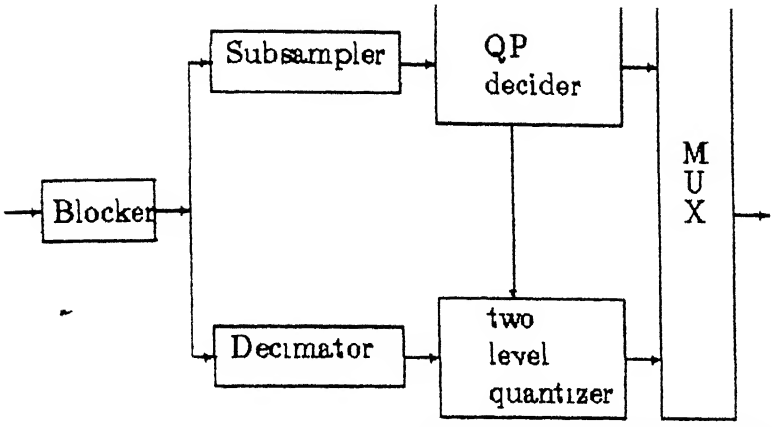


Figure 3.1: Modified BTC

subsampled blocks compared to the algorithm used in the block truncation coding. Finally the subsampled pixels are quantized using two level quantizer and then a bit plane is formed which along with the reconstruction levels are transmitted.

The decoder of the modified BTC recieves the bitplane along with reconstruction levels. Then it reconstructs the quantized pixels and interpolates the pixels eliminated by the subsample process, synthesizes and generates an output image.

The block diagram of Encoder of modified BTC is shown in Figure 3.1.

The encoder of modified block truncation coder consists of blocker, subsampler, decimator, quantizer parameter decider and the two level quantizer. The blocker divides the image into several 8×8. blocks. If the original image data is $OX(n)$, the blocker divides the original image data $OX(n)$ into blocks $X(n)$ with size 8×8 These blocks are then sent into the subsampler. Let $X(n)$ and $SX(n)$ denote the input and output data of subsampler, respectively. Then the function performed by subsampler is as follows.

$$SX(n) = X(k \times n)$$

where, k is a positive integer.

The array below shows the function performed by subsampler.

$$X = \begin{bmatrix} \boxed{x_{00}} & x_{01} & x_{02} & \boxed{x_{03}} & x_{04} & x_{05} & \boxed{x_{06}} & x_{07} \\ x_{10} & \boxed{x_{11}} & x_{12} & x_{13} & \boxed{x_{14}} & x_{15} & x_{16} & \boxed{x_{17}} \\ x_{20} & x_{21} & \boxed{x_{22}} & x_{23} & x_{24} & \boxed{x_{25}} & x_{26} & x_{27} \\ \boxed{x_{30}} & x_{31} & x_{32} & \boxed{x_{33}} & x_{34} & x_{35} & \boxed{x_{36}} & x_{37} \\ x_{40} & \boxed{x_{41}} & x_{42} & x_{43} & \boxed{x_{44}} & x_{45} & x_{46} & \boxed{x_{47}} \\ x_{50} & x_{51} & \boxed{x_{52}} & x_{53} & x_{54} & \boxed{x_{55}} & x_{56} & x_{57} \\ \boxed{x_{60}} & x_{61} & x_{62} & \boxed{x_{63}} & x_{64} & x_{65} & \boxed{x_{66}} & x_{67} \\ x_{70} & \boxed{x_{71}} & x_{72} & x_{73} & \boxed{x_{74}} & x_{75} & x_{76} & \boxed{x_{77}} \end{bmatrix}$$

$$SX = \begin{bmatrix} x_{00} & x_{03} & x_{06} \ldots x_{77} \end{bmatrix}$$

X is the input to the subsampler and SX is the output of the subsampler with k set to 3 The framed pixels are selected by the subsampler.

The subsampler output is fed into the quantizer parameter decider which generates the threshold QT and the two reconstruction levels. In the BTC the original blocks are used to generate QT and the two reconstruction levels, but in modified BTC the subsampled blocks are used to generate QT and the two reconstruction levels. This modification is done so as to increase the computation speed.

The parameters QT and the two reconstruction levels RL(0) and RL(1) are defined by the following equations

$$QT = \frac{1}{\lceil \frac{M}{k} \rceil} \sum_{0 \le i < \lceil \frac{M}{k} \rceil} SX(i)$$

which means it is the average value of the subsampled pixels.

$$RL(0) = \frac{1}{\sum_{\lceil \frac{M}{k} \rceil} \delta_{SX<QT}} \sum_{0 \le i < \lceil \frac{M}{k} \rceil} \delta_{SX(i)<QT} SX(i),$$

$$RL(1) = \frac{1}{\sum_{\lceil \frac{M}{k} \rceil} \delta_{SX \ge QT}} \sum_{0 \le i < \lceil \frac{M}{k} \rceil} \delta_{SX(i) \ge QT} SX(i)$$

where k is the value chosen at the subsampler and

$$\delta_f = 1$$

when f is true, otherwise

$$\delta_f = 0$$

In otherwords RL(0) is the average value of subsampled pixels which are less than QT, and RL(1) is the average of subsampled pixels which are greater than or equal to QT

The function of decimator is similar to subsample function of subsampler. The two level quantizer quantizes the decimated image DX(n), using the quantizer threshold QT. After quantization a bit plane is formed. The bit plane B(n) output by the two level quantizer is defined by

$$B(n) = \delta_{DX(n) \ge QT}$$

where $\delta_f = 1$ when f is true, otherwise $\delta_f = 0$

The bit plane is then multiplexed with the reconstruction levels RL(0) and RL(1) and sent to the modified BTC decoder.

At the decoder the bitplane is converted back to decimated pixels using the two reconstruction levels, i.e., the inverse quantizer performs an inverse function of two

level quantizer. Because the image pixels reconstructed by the inverse quantizer are decimated pixels, the pixels eliminated by the decimator must be recovered by the interpolator. Straight line interpolation was used for implementation. Finally the recovered image blocks are synthesized as a single frame to form an output image

3.3 Implementation of modified BTC

The modified BTC was implemented using 32×32 eye image, on a PC/AT 386 machine using C language programming

First the image was divided into 8×8 non overlapping blocks. To illustrate the steps involved in modified BTC the first block of the image is reproduced below

$$\begin{bmatrix} 118 & 113 & 85 & 101 & 161 & 151 & 151 & 163 \\ 118 & 85 & 101 & 151 & 140 & 144 & 157 & 162 \\ 89 & 90 & 141 & 146 & 140 & 138 & 145 & 166 \\ 84 & 141 & 158 & 134 & 138 & 138 & 143 & 155 \\ 129 & 161 & 140 & 144 & 147 & 137 & 140 & 156 \\ 171 & 145 & 152 & 160 & 153 & 141 & 138 & 157 \\ 154 & 147 & 162 & 170 & 151 & 133 & 129 & 155 \\ 150 & 161 & 174 & 180 & 143 & 133 & 138 & 168 \end{bmatrix}.$$

Next step is to subsample each block Subsampling was carried out using k = 3. Reproduced below is the first block of the image eye after subsampling

$$[118, 101, 151, 85, 140, 162, 141, 138, 84, 134, 143,$$
$$161, 147, 156, 152, 141, 154, 170, 129, 161, 143, 168]$$

Threshold QT = $146.61 \simeq 147$

The two reconstruction levels are

$$RL(0) = 124.75 \simeq 125,$$
$$RL(1) = 158\ 19 \simeq 158,$$

and the bit plane

$$[0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 1, 1, 1, 1, 0, 1, 1, 0, 1, 0, 1].$$

The bit plane alongwith the two reconstruction levels were transmitted.

At the reciever, the recieved bit plane was reconstructed using the two reconstruction levels as shown below..

$$[125, 125, 158, 125, 125, 158, 125, 125, 125, 125, 125,$$
$$158, 158, 158, 158, 125, 158, 158, 125, 158, 125, 158]$$

Next step is to interpolate the left out values during decimation. A simple straight line interpolation was used to interpolate the left out values.

Shown below is the first block after interpolation.

$$\begin{bmatrix} 125 & 125 & 125 & 125 & 136 & 147 & 158 & 147 \\ 136 & 125 & 125 & 125 & 125 & 136 & 147 & 158 \\ 147 & 136 & 125 & 125 & 125 & 125 & 125 & 125 \\ 125 & 125 & 125 & 125 & 125 & 125 & 125 & 136 \\ 147 & 158 & 158 & 158 & 158 & 158 & 158 & 158 \\ 158 & 158 & 158 & 147 & 136 & 125 & 136 & 147 \\ 158 & 158 & 158 & 158 & 147 & 136 & 125 & 136 \\ 147 & 158 & 147 & 136 & 125 & 136 & 147 & 158 \end{bmatrix}$$

Finally all the blocks were reorganized into a single frame and the reconstructed image was displayed.

3.4 SBMPC SYSTEM

The block diagram of SBMPC encoder is in Figure below. The input image is first partitioned into image blocks by the blocker. Then these image blocks are sent to the PCM and the motion detector. When the image sequence contains a scene change or some objects in the image sequence move quickly, the image would contain lot of information and must be detected. The function of the motion detecter is to detect for any movement in the present frame compared to the previous frame. To detect the movement DPCM technique is used.

The motion detector compares a block in the present frame with the corresponding block in the previous frame and detects whether there was any motion The multiresolution sampler gets the image blocks sent by the PCM along with the information regarding the movement. Then the blocks are divided into several activity classes

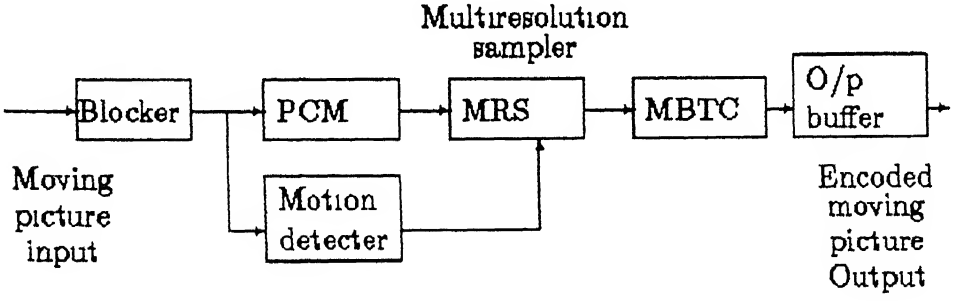


Figure 3.2: Block diagram of SBMPC encoder

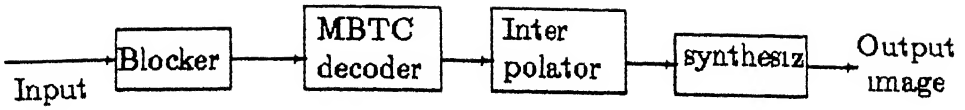


Figure 3 3· Block diagram of SBMPC decoder

The activity of an image block is defined by

$$ACT = RL(0) - RL(1)$$

where, $RL(0)$ and $RL(1)$ are defined by the equations for reconstruction levels in the Modified BTC.

Depending upon the activity, the multiresolution sampler changes the subsampling rate in the modified BTC. The modified BTC subsamples each block with different subsampling rate depending upon the activity. When the objects in the video are moving, it is more important to trace the actions of the objects fast and clearly, than to describe the contents of these objects in detail. Because the image blocks which contain a part of moving objects usually have higher activity values than the blocks which contains a part of the backround, the multiresolution sampler should ensure that the modified BTC codes the higher activity blocks with higher sampling rate and lower activity blocks with lower sampling rate.

Hence the modified BTC codes different blocks with different subsampling rate and sends the output data which consists of bitplane, reconstruction levels and the information regarding the subsampling rate for each block to the SBMPC decoder

The block diagram of SBMPC decoder is as shown in Fig. 3 3

The decoder recieves the bitplane, two reconstruction levels and the information regarding the subsampling rate for each block. The bitplanes of each block is reconstructed using two reconstruction levels. i.e., zeroe's in the bitplane are replaced by $RL(0)$'s and one's in the bitplane are replaced by $RL(1)$'s.

After reconstructing the bitplane, the decoder checks for the information, at what rate the block was subsampled at the encoder. Depending upon the subsampling rate used at the encoder, the blocks are interpolated suitably

3.5 Implementation

The SBMPC was implemented on PC/AT 386 using 'C' language programming. Since the SBMPC system is for moving pictures, several frames of continuous sequence was necessary. Fifteen frames of a sequence of a person moving his head from left to right was used These frames were initially stored in the computer as Frame1, Frame2, . ., Frame15. Frame size of 64×64 was used.

The incoming picture frames are divided into several nonoverlapping blocks. Block size of 8×8 is used. Since the frame size is 64×64 each frame will be divided into 64 blocks of block size 8×8 After dividing the picture frame into blocks, the program checks whether a particular block belongs to frame1 or not. If the block is from frame1, modified BTC with k = 1 is applied to that block, and the corresponding bit plane and the two reconstruction levels are transmitted. All the blocks of frame1 are similarly coded and transmitted.

However, if the program finds that a particular block is not from frame1, it has to first of all detect whether there was any movement in that block compared to the corresponding block in the previous frame. If there is a moevment or change, the program has to also find the amount of change detected. For this purpose, the blocks are divided into three activity classes. They are Activity class x, Activity class y, Activity class z

The flow charts for the implementation of SBMPC encoder is shown in the following page.

To detect the movement in a particular block, the difference between that block and the corresponding block in the previous frame is calculated. Then the activity of that block is given by $ACT = RL(0) - RL(1)$. The blocks were divided into three activity classes as mentioned earlier. Activity x, very little or no change

$ACT\ x \leq 10$

Activity y, moderate change

$10 < ACT\ y \leq 25$

Activity z =, large change

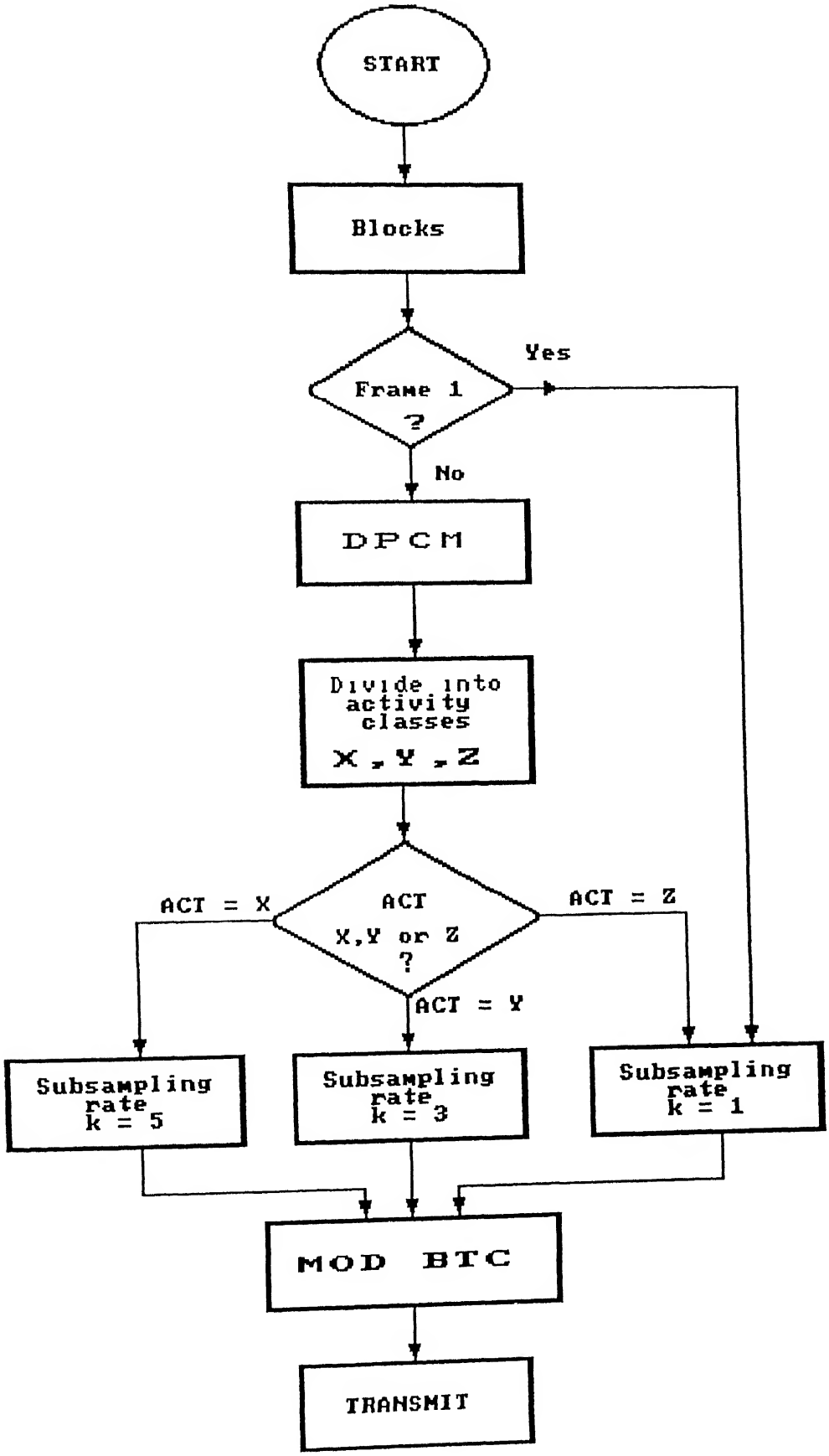
$ACT\ z > 25$.

For all the blocks with activity x, modified BTC with $k = 5$ is applied and for all the blocks with activity y, modified BTC with $k = 3$ is applied and finally for all the blocks with activity z, modified BTC with $k = 1$ is applied.

The modified block truncation coder generates corresponding bit plane and reconstruction levels for each block which are transmitted

The decoder recieves the bitplane, two reconstruction levels and the information regarding the subsampling rate The first thing that the decoder does is to reconstruct the bitplane by replacing 0s in the bitplane by $RL(0)$s and 1s in the bitplane by $RL(1)$s. Once the reconstruction is completed the decoder has to interpolate the left out pixels

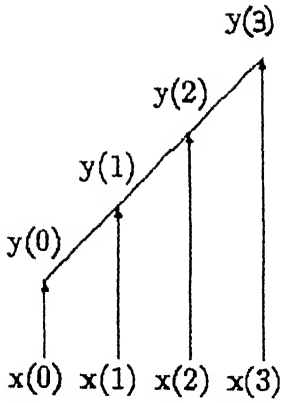
SBMPC Transmitter



during the decimation operation. Since the subsampling rate is different for different blocks, the interpolator should adapt itself according to the subsampling rate.

To simplify the interpolation operation, what was done is, each block was checked for its subsampling rate which is known by the value of k. If k is found to be 3, three zeroes were padded inbetween all the values of reconstructed block. If k is found to be 5, five zeroes were padded inbetween all the values of reconstructed block. However, if k is found to be 1, no padding necessary.

Finally using a simple straight line interpolation the left out values are interpolated and is illustrated in the next page.



$y(0)$ and $y(3)$ are known. $y(1)$ and $y(2)$ are unknown and are to be interpolated.

$$
\begin{aligned}
y(3) &= 3m + c. \\
y(0) &= c. \\
m &= \frac{y(3) - y(0)}{3}. \\
y(i) &= \left(\frac{y(3) - y(0)}{3}\right) x + y(0) \\
y(1) &= \left(\frac{y(3) - y(0)}{3}\right) + y(0) = \frac{y(3) + 2y(0)}{3} \\
y(2) &= \left(\frac{y(3) - y(0)}{3}\right) 2 + y(0) = \frac{2y(3) + y(0)}{3}
\end{aligned}
$$

All the fifteen frames were encoded by the SBMPC and the decoded image was displayed, and the compression ratio was calculated The quality of the decoded image were satisfactory, but can still be improved by using better interpolation technique and may be multilevel quantization.

Chapter 4

Conclusions

Software based motion picture coder is computationally high speed algorithm with high compression ratio. The main advantage of SBMPC is that it is easy to implement. The heart of the system is the modified BTC which employs subsampling process which can both accelerate the computations and achieve higher compression ratio. Furthermore, a simplified algorithm is used for computing the quantizer threshold and the two reconstruction levels for each subsampled block

The quality of decoded image is not of high quality, mainly because in modified BTC two level quantizer are used at the coder and a simple straight line interpolator is used at the decoder. To improve the quality of the decoded image multilevel quantization can be used at the coder and at the decoder still better interpolation technique can be employed Median filter can be used as a interpolater. The main advantage is it gives good results around sharp edges.

The compression ratio is given by

$$Compression ratio = \frac{Original\ \ data\ \ size}{Encoded\ \ data\ \ size}$$

For SBMPC compression ratio for all the fifteen frames were calculated and it was found that on an average the compression ratio was found to be 25. However, the compression ratio can be changed depending upon the threshold set for the activity classes. A trade off between compression ratio and quality of the decoded image can be arrived at.

When the computation power of PC is inceased, the performance of SBMPC can be improved in several ways. First, some simple and fast motion detection algorithm can be used to remove the temporal redundancy further. Secondly, some fast transform algorithm can be used to remove the spatial redundancy. Third, entropy coding can be introduced into SBMPC system to encode data more compactly. Finally, the frame rate can be increased and the frame size can be enlarged

REFERENCES

[1] Ho-Chao Huang and ja-ling Wu. Novel real-time software based video coding algorithms. *IEEE trans. on consumer electronics*, pages 570-580, August 1993

[2] Jain. A.K Fundementals of digital image processing.

[3] Lim. Two dimensional signal and image processing.

[4] Rao.K.R. and Yip.P. Discrete cosine transform.

[5] Arun N. Netravali. Picture coding· A Review. *Proc. IEEE*, Pages 366-407, March 1980.

[6] Anil K Jain. Image data compression. A review *Proc. IEEE*, Pages 349-389, March 1981.

[7] Raj Natarajan, T On Interframe Transform coding. *IEEE Trans. on communication*, Pages 1323-1329, Nov 1977.

EE-1594-D-SEB-SOF